基于 XML 刻面构件描述与检索算法研究①

Research on the Representation and Retrieval Algorithm of XML – based Facet Component

鲁大营 曹宝香 （曲阜师范大学计算机科学学院 山东日照 276826）
王 华 （西安科技大学通信与信息工程学院 陕西 西安 710054）

摘 要： 随着构件库的规模的扩大以及构件的复杂性增加和语义性丰富，单一的软构件描述以及在此基础上的检索算法将很难再胜任整个构件库的检索任务。在基于刻面描述的软构件的基础上引入 XML 技术，来全面描述构件的静态属性、接口行为、组织分类和资源位置等信息，方便构件检索过程中信息的交换与处理，也便于与其他系统的互操作。并就此提出了一种新的构件检索方案，就算法的时间开销和空间开销进行分析，并利用试验数据说明算法的可行性和执行效率。

关键词： 刻面 XML 软件复用 构件描述 检索 匹配

随着软件复用实践的深入以及新信息技术的涌现，针对软构件的描述与检索的研究也不断的取得进展。针对构件的不同描述方法，也涌现出不同的检索策略，例如，有根据构件刻面描述的特点提出的一种树包含的构件检索方法[2][3]，有针对构件的行为表示提出的一种基于构件行为采样的检索方法[7]，还有基于正文的分类和检索、基于词法描述符的分类和检索、基于规约的分类和检索[1]等等。随着领域知识的不断扩充及构件复杂性增加和语义性丰富，上面提到的几种针对不同构件表示模型的构件检索方案虽然在查询到所需的构件方面解决一定的问题，但是在查全率、查准率、查询代价方面呈现出一定的不足，且很难在异质构件库间达到构件共享。因此，本文在研究软件构件查询具体特点的基础上，借鉴刻面描述的构件查询匹配模型及关键字的检索方法，，提出一种能够兼具较高的构件查全率、查准率和查询效率的软件构件组合检索算法。

1 基于 XML 的刻面构件描述

Internet 上不断扩展的网络构件资源为网络服务和进行程序挖掘提供了基础[5]。尽管 Internet 上出现了多个专门的构件库，如 Alphaworks、ComponentPlanet、Componentsource、Flashline 等，提供了多种现成的可用构件，但由于这些构件库在组织结构、构件描述和访问方式上各不相同，造成构件选择、搜索、获取以及分析、组装等构件处理活动的困难。要实现软件构件的重用，首先要解决不同构件库之间构件访问的一致性问题[4]。构件描述与组织的任务就是通过建立分布式构件目录信息库，为不同的构件库提供统一的浏览和搜索接口；同时在构件库与目录库之间，通过建立多个代理完成构件信息的注册、取消和动态更新，屏蔽多种构件库之间的差异，提供组织良好的构件资源。

在[6]中作者指出软件重用的关键在于能否清晰无二义性的描述我们需要什么样的构件，或者能否快速找到已有的符合要求的构件。为此，在基于刻面描述的软构件的基础上引入 XML 技术，来全面描述构件的静态属性、接口行为、组织分类和资源位置等信息，方便构件检索过程中信息的交换与处理，也便于与其他系统的互操作。且这种描述方法可以描述现有的主流构件，如：Javabean、EJB、DCOM 构件、COM 构件、CORBA 构件等，还可以根据构件的发展对构件描述的主要内容进行扩充，实现构件属性的动态定义。

① 基金项目：山东省自然科学基金项目（Y2003G01）

为了满足构件描述独立于特定构件技术和易于扩展的要求，在确定了构件描述符中各部分的内容后，我们选用扩展标记语言 XML 来表示构件描述符。XML 是 W3C 组织提出的标准，是一种与平台无关的结构化信息描述语言，是目前 Web 中最通用的数据描述格式。构件描述符中的各个描述项可以通过定义相应的 XML 标签来表示，这样每个构件的描述就成为一个 XML 文档，构件的搜索、访问等操作就可以转换为对 XML 文档的读写和检索处理。为了规范用来表示构件描述符的 XML 标签，我们设计了相应的 DTD（Document Type Definition），该 DTD 可用来验证构件描述文档的合法性，其构件描述文档的 DTD 部分片断如图 1 所示。

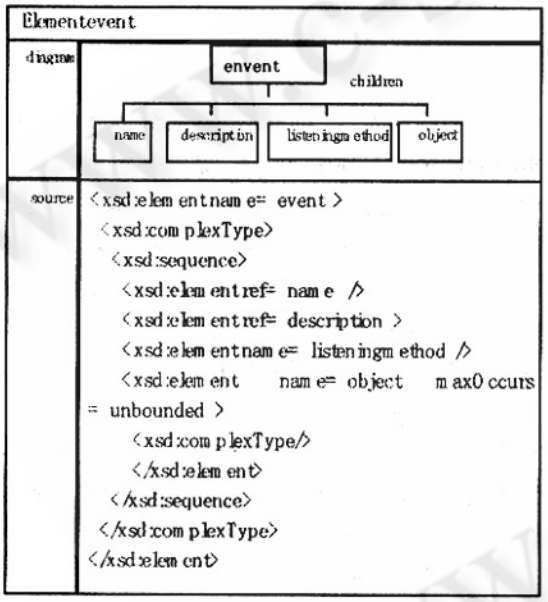

图 1　构件描述文档的部分片断

为了能给用户提供统一的构件查询接口，屏蔽多个构件库之间的不兼容性，我们选用源代码开放的专用 XML 数据库 dbXML 来存储和组织构件描述文档，并利用 dbXML 的 Java API 设计了构件浏览和检索界面，对外提供构件库中的构件目录信息服务。dbXML 数据库中集合相当于文件系统中的目录或文件夹，所有用户集合都在一个称为 root 的根集合下创建，在 root 以下，集合之间可以多层嵌套，在集合中存储 XML 文档时，文档首先被索引和压缩，以提供存储和检索的效率。为了在 dbXML 中存储构件描述文档，我们首先在 root 集合下创建 ComPonent 集合，然后通过调用 dbXML 提供的 XML:DB API，访问 XML 数据库，在 Component 集合下进行构件描述文档的创建、修改、删除等操作。如图 2 为构件描述文档在 dbXML 中的存储算法。

```
//获取数据库实例
String driver = " org. dbxml. client. xmldb. DatabaseImPl" ;
Classc = Class. forName( driver) ;
Database database = ( Database) c. newInstance( ) ;
DatabaseManager. registerDatabase( database) ;
//获取 component 集合实例
col = DatabaseManager. getCollection ( " xmldb: dbxml:/// db/
component" ) ;
//创建 XMl 文档
string data = readFileFromDisk ( filename) ;//读入构件描述文档
内容
XMLResource document = ( XMLResource ) col. createResource
( null," XMLResource" ) ;
document. setContent( data) ;
col_storeResource( dncument) ;
```

图 2　构件描述文档在 dbXML 中的存储算法
dbXML – Insert Algorithm

2　基于 XML 刻面构件描述

在借鉴构件刻面描述检索和基于关键字的检索策略的优点的基础上，提出一种基于这两种检索策略组合的检索方法，以提高构件的查全率、查准率和查询效率。图 3 为构件检索过程。

用户根据需求向用户接口（User API）提交检索请求，生成规范的 XML – QL 语句，通过分析应用程序（Analytical – prg）对查询模式进行优化和纠错，生成高效的 XML – Tree，并进行功能匹配，通过匹配算法得出每个构件与用户需求的匹配度，通过匹配度来选择出合适的构件。

我们可以把用户需求的功能模块作为构件的功能描述，这样就可以用一个树来表示构件的功能描述，构件功能之间的比较可以使用树的匹配算法。虽然树的精确匹配算法比较成熟，但不适合于构件功能之间的比较。这时我们只要对树的结点之间定义简约规则就

可以进行构件需求与构件功能之间的模糊匹配。根据用户需求建立功能需求树，且只有一个根结点，还要构

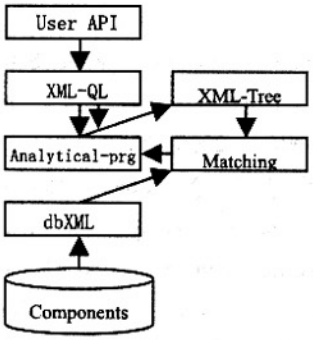

图 3　构件检索过程

建与其相对应的所有方案树，它们的根结点是相同的，并且功能需求树中的所对应的结点的子结点都必须包含在方案树中，这样就可以对每一个方案树与功能需求树匹配度进行计算，按照匹配度的大小进行排列，输出解决方案。

在同一领域中进行构件功能之间的比较（匹配度），则以刻面匹配分类组织以结点为单位进行，即构件每个刻面匹配度和除以构件刻面数。计算公式如下：

$$M(c1,c2)=\sum_{i=1}^{n} l(Ti,Qi)/n$$

$$l(Ti,Qi)=/\ Ti\cap Qi\ ///\ Ti\cup Qi\ /$$

其中 $l(Ti,Qi)$ 为构件 T 与构件 Q 刻面 i 的匹配度，且 Ti 与 Qi 是构件的独立刻面，c1 与 c2 分别代表的是构件的刻面的列表。通过对匹配度的计算可以在一定程度上满足查询用户对构件的检索需求。

我们可以在基于 XML 的构件描述信息的基础上，通过刻面分类对构件进行组织，建立构件目录信息库（dbXML），为多个结构各异的构件库提供统一的访问接口。图 4 为构件描述文档的检索算法。

3　算法测试及相关的实验数据

为了测试和验证我们所提出的构件检索策略的性能，我们设计了几种测试数据，检验构件检索策略的时间复杂度（图 5）和检索效率（图 6）。

```
//获取数据库实例
String driver = "org. dbxml. client. xmldb. DatabaseImPl";
Classc = Class. forName(driver);
Database database = (Database)c. newInstance();
DatabaseManager. registerDatabase(database);
//获取 component 集合实例
col = DatabaseManager. getCollection("xmldb:dbxml:/// db/root/
component");
string xpath = XpathExpression;//得到搜索表达式
//通过集合获取 Xpath 查询服务
    XPathQueryservice service =
    (XPathQueryservice)col. getService("XPathQueryService","1.
0");
//执行查询，返回结果集合
    ResourceIterator results = service. query(xpath);
```

图 4　构件描述文档的检索算法

构件数量	20	40	80	160	240	320
算法执行时间	112	136	155	267	553	762

图 5　小容量构件库中的检索时间

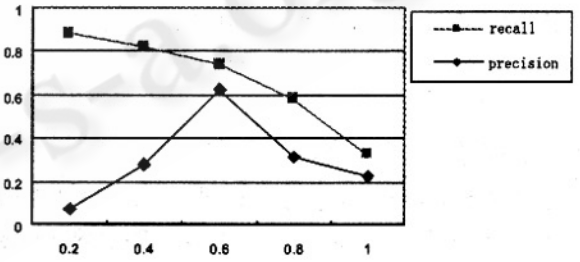

图 6　此算法的检索效率参数值

4　总结和展望

本文针对构件进行刻面分类组织，并引入 XML 技术对构件刻面描述其静态属性、接口行为、组织分类和资源位置等信息，方便构件检索过程中信息的交换与处理，也便于与其他系统的互操作。基于词法描述符的方法是当前构件库分类检索方法中研究得最深入、应用得最普遍，同时也是在检索代价、复杂性和检索质

量这三者之间最为均衡的方法。因此本文采用的结合此两种检索策略的优点，提出了一种高效的构件检索方案，并分析测试了该检索方案的时间开销和检索代价，实验表明其虽然实现了较高的检索效率，但检索时间复杂度较高。因此，进一步的研究工作主要是研究对刻面赋以权值来缩短检索时间以及引入同义词库对构件的功能信息进行扩大并结合相关反馈技术来提高检索质量。

参考文献

1 MA Liang, SUN Jia - su. Component Retrieval Based on Specification Matching [J]. INI - MICRO SYSTEM, 2002, 23(10): 1153 - 1157.

2 WANG Yuan - Feng, XUE Yun - Jiao, ZHANG Yong, et al. A Matching Model for Software Component Classified in Faceted Scheme [J]. Journal of Software, 2003, 14(3): 401 - 408.

3 JIA Xiao - Hui, CHEN De - Hua, YAN Mei, et al. Research on Matching Model and Algorithm for Faceted - Based Software Component Query [J]. JOURNAL OF COMPUTER RESEARCH AND DEVELOPMENT, 2004, 41(10): 1634 - 1638.

4 CHANGJi - chuan, LI Ke - qin, GUO Lifeng, et al. Representing and Retrieving Reusable Software Components in JB (Jadebird) System [J]. ACTA ELECTRONICA SINICA, 2000, 28(8): 20 - 23.

5 Xiao Hang Wang, Da Qing Zhang, Tao Gu, et al. Ontology based context modeling and reasoning using OWL. Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops (PERCOMW 2004). 2004: 18 - 22.

6 Rogaway. ComponentPlanet, 2004/322. http://www.ComponentPlanet.com

7 Hu J, Yu XF, Zhang Y, Wang LZ, Li XD, Zheng GL. Checking component - based designs for scenario - based specification [J]. Chinese Journal of Computers, 2006, 29(4): 513 - 525.